Median Filtering Forensics based on Convolutional Neural Network and Local Optimal Oriented Patterns

Ali Ahmad Aminu
Computer Science Department
Nile University of Nigeria, Abuja

Nwojo Nnanna Agwu
Computer Science Department
Nile University of Nigeria, Abuja

Nzurumike Obianuju
Computer Science Department
Nile University of Nigeria, Abuja

ABSTRACT

Median filtering forensics has been an active area of research in recent times due to its inherent nature of preserving visual artifacts. To create a convincing image manipulation, Forgers often apply median filtering to destroy statistical traces introduced during image manipulation, hence, median filtering detection has gained wide attention from digital image forensics researchers recently. While many median filtering forensics methods have been developed, the performance of these approaches degrades in low–resolution images compressed with low compression quality factor. This study presents a novel method for median filtering detection based on Local Optimal Oriented Pattern (LOOP) and Convolutional Neural Network (CNN). Here, we employed LOOP, Local textural descriptors of images which can better capture textural variation introduced during image manipulation, as the input of the proposed model. To test the performance of the proposed method, we evaluate its performance using composite datasets formed from five publicly available image datasets. Experimental results demonstrate that the proposed method outperforms some exiting state of the art and could be potentially used to enhance median filtering detection in highly compressed low-resolution images.

General Terms

Median Filtering Forensics, Deep Learning

Keywords

Median filtering Detection, Convolutional Neural Network (CNN), Local Optimal Oriented Pattern (LOOP), low-resolution image,

1. INTRODUCTION

With the advent of high-performance digital image editing software such as Photoshop and GIMP, it has become relatively easy to manipulate digital image information. Anyone with rudimentary image editing skills may tamper with the original image information. Hence, it becomes necessary to verify the authenticity of digital information before using it to make vital decisions. Consequently, the field of digital image forensics has attracted the attention of many researchers in the past decade.

Active and passive or blind methods have been the two major forensics approaches for authenticating images [1]. The active approach makes use of watermarks and digital signature, which must be inserted in the image at the time of capture, as a means of verifying whether it has tampered with or not. Although the active approach is effective, the need for a special imaging device has narrowed its applications in digital image forensics [2]. As a result, researchers have focused more on developing the blind methods which operate based on the fact that image tampering operations inevitably alter the underlying statistical properties of an image and this can be detected by detecting changes in the natural statistical properties of an image [3].

However, to create a convincing image tampering, forgers often use content preserving manipulations such as JPEG compression, contrast enhancement, and median filtering to destroy statistical change introduced by image manipulation operations [4]. Among the different content preserving manipulations, median filtering is the most widely used for hiding image manipulation due to its smoothing properties and noise removal capabilities [5]. Thus, median filtering detection has gained the wide attention of researchers, and several median filtering detection methods based on hand-designed features [5], [6] and CNN [7], [8], [9] have been proposed.

Although these methods have improved the state of the arts, their performance degrades in low-resolution images compressed with low-quality factor. This paper presents a novel method for median filtering detection based on Local Optimal Oriented Pattern (LOOP) and Convolutional Neural Network (CNN). The proposed method utilizes the high texture description strength of LOOP, which is capable of encoding image pixel variation induced by tampering operations and the excellent learning abilities of Convolutional Neural Network (CNN) to build a general framework for median filtering detection. Unlike existing techniques that utilize the actual image as input to their CNN model, we employed LOOP representation of images as the input of our proposed CNN design due to their capabilities of suppressing image content, thus, revealing manipulation traces [10] and the fact that image manipulation traces are more pronounced in LOOP image than the original image.

For a thorough assessment of the proposed method, its performance has been compared with existing hand-crafted based approaches and deep learning approaches from the literature under different experimental setup on composite database formed from five publicly available databases; UCID, BossBase, MS COCO, NRCS Photo Gallery and Dresden image database. Experimental results show that the proposed method outperforms existing methods and could be potentially used to enhance median filtering detection in low-resolution images with low compression quality factors.

The contribution of this paper is two-fold (1) novel CNN architecture that uses LOOP representation of images as input, (2) enhanced performance in median filtering detection.

The rest of this paper is organized as follows: Section 2 gives the related work. Section 3 discusses the proposed method. Section 4 presents the experiments and discusses the experimental results. Section 5 concludes the paper.

2. RELATED WORK

Median filtering is a widely used de-noising and smoothing post-processing operation with a wide interest in digital image forensics examination due to its nonlinearity and edgepreserving capabilities [4]. Since most image forensics methods depend on one kind of linearity assumption or the other, these properties are utilized by anti-forensics methods to interfere with or hide the subtle traces of previous image manipulations, thus, decreasing the reliability of the forensics methods. Therefore, the detection of median filtering is a crucial step in revealing the processing history of an image under investigation.

A handful of studies have been proposed for detecting median filtering. Earlier median filtering detection approaches used handcrafted features based methods [11], where image statistics are analyzed to capture the artifacts of median filtering. In [12], Yuan constructed a 44-dimensional feature set, named Median Filtering Forensics (MFFs) by exploiting the relationship between different pixels of small window. MFFs efficiently detect median filtering in uncompressed images but fails in median filtered images that have undergo post JPEG compression. In another study, Niu *et al* [13] proposed a novel 128 feature set, named Local Difference Descriptors (LDD) formed by combining two subsets based on Local Binary Pattern (LBP) and Pixel Difference Matrix (PDM). Their experiments show that the proposed method can efficiently detect median filtering in low-resolution images under JPEG compression attacks. However, the feature set proposed in [13] was still large leading to a high cost of computation. Recently, Gupta [5] has exploited the statistics of person parameter k, to capture the fingerprints of median filtering from which they constructed 23 new dimensional feature sets for median filtering detection. Results from their experiments show that the proposed method outperformed existing handcrafted features based detectors and CNN based detectors. Furthermore, in [14], a 19 feature set is constructed by studying image skewness and kurtosis histogram for detecting global median filtering. Extensive experiments on six benchmark datasets demonstrate that the proposed method is robust in detecting median filtering in low-resolution images under JPEG compression.

Recently, deep learning approaches particularly CNN, have also been successfully used for developing median filtering detection methods with a better performance when compared to the handcrafted features based approaches. In [8], Chen proposed the first CNN based median filtering detection approach that can automatically capture median filtering traces from input images. In another similar work [7], Tang *et al* also proposed a CNN based median filtering detection scheme for low-resolution images compressed with low-quality factors. Tang used nearest-neighbor interpolation to up-sample small-sized images before feeding them to the CNN. The results of their experiments indicate that their method significantly improves median filtering detection in low-resolution images with low-quality factors. Furthermore, Yu [9] proposed a CNN based multiple residual learning for detecting median filtering in highly compressed low-resolution images. Yu utilized multiple high pass filters to initialize the weights of the initial layers of their network to obtain discriminative residuals capable of characterizing median filtering traces in various aspects. Experimental results show that their method outperformed state of the art methods in detecting median filtering in compressed images irrespective of the image resolution.

3. PROPOSED METHOD

In this section, we present an overview (figure 1) of the proposed approach for detecting median filtering in digital images. The proposed method utilizes the high texture description strength of LOOP, which is capable of encoding image pixel variation introduced by tampering operations and the excellent learning abilities of Convolutional Neural Network (CNN) to build a framework for median filtering detection. Unlike existing techniques that utilize the actual image as input to their CNN model, we employed the LOOP representation of images as the input of our proposed CNN due to their rotation invariance and the fact that image manipulation traces are more pronounced in LOOP image than in the original image [15].

3.1 Local Optimal Oriented Pattern (LOOP

LOOP is a textural binary descriptor recently proposed by Chakraborti et *a*l [10], mostly used in the classification of texture. Texture in this context refers to grayscale or color pixel intensities of an image. LOOP is an enhancement of Local Binary Pattern (LBP) [16], and Local Directional Pattern (LDP) [17] and their variants which have recorded much success in image classification tasks including image tampering detection. LOOP improved on existing local descriptors by adding rotation invariance into the main formulation of local descriptors, hence overcoming their limitation, decreasing computational time, and increasing overall accuracy [10]. Inspired by this, we employed LOOP representations of images instead of original images as the input of the proposed model.

3.2 LOOP Computation

Given an image *I*, LOOP representation can be obtained using equation (1) and (2). Suppose i_c represent the intensity of the image *I* at pixel (x_c, y_c) and i_n (n = 0, 1… 7) is the intensity of a pixel in the 3×3 neighborhood of (x_c, y_c) ignoring the center pixel i_c . The 8 Kirsch masks are oriented in the direction of the 8 neighboring pixels i_n (n = 0, 1, 2… 7), giving a measure of the strength of intensity variation in those directions respectively.

Suppose m_n denotes the responses of the 8 Kirsch masks corresponding to pixels with intensity i_n (n = 0, 1, 2… 7). An exponential weight w_n (a digit between 0 and 7) is assigned to each of these pixels according to the rank of the magnitude of m_n among the 8 Kirsch masks activations.

Then, the LOOP value for the pixel$(x_c, y_c))$ can be computed as:

$$Loop\left(x_c, y_c\right) = \sum_{n=0}^{7} f\left(i_n - i_c\right).2^{w_n} \qquad (1)$$

Where

$$f(x) = \begin{cases} 1, & if \quad x \geq 0 \\ 0, & otherwise \end{cases} \qquad (2)$$

The LOOP image is formed by accumulating the occurrence of the LOOP values over the entire image, i.e. each image is represented as a collection of LOOP values obtained from equation (1). Each of i_n neighboring pixel oriented in the direction of the response of the 8 kirsch masks is compare with the central pixel i_c which evaluates to either 0 or 1 according to equation (2) forming an 8 digit binary number. Weights are then assigned to each of these binary numbers according to the rank of the magnitude of m_n among the output of the 8 Kirsch masks. The resulting binary number is then used as a label for that central pixel. For convenience, the binary number is converted to decimal. Repeating this procedure for the entire pixels of an image, the LOOP image is computed having the same size as the original image. The resulting LOOP images are used as input to our proposed model.

3.3 The Proposed CNN Architecture

Figure 3 illustrates the overall architecture of the proposed CNN for median filtering detection. The proposed CNN contains 11 layers, the first 9 are convolution layers and the last 2 are fully connected layers. Batch Normalization operations are applied after each convolution operation followed by non-linear mapping and pooling operations in some of the layers. Specifically, non-linear mapping and pooling operations are disabled in layers (1-3). Layers (4-10) used tanh non-linearity after batch normalization followed by average pooling.The output of the last FC layer is fed to a softmax activation which transforms feature vectors to output probabilities for each class. In the proposed CNN design, the input layer is a 256×256 and 128 grayscale LOOP images, and network parameters such as the number of filters, their size, and initial values are arrived at experimentally. Below, we describe some of the features of the new network architecture.

3.3.1 Weight Initialization

CNN with many layers are usually difficult to train due to their huge amount of parameters and the fact that their loss function is non-convex [18], however, their performance can be optimized by proper network initialization. In the proposed CNN design, we experimented with different weight initialization schemes namely Xavier initialization with normal bias and zero bias, He-normal with normal bias and zero bias, and lastly truncated normal with normal and zero bias. Experimenting with all methods of initialization leads to the realization that our model performs best when the weights are initialized using Xavier initialization and biases are initialized with zeros in all convolutional layers. Unlike other related studies such as [19-20], we did not constrain the parameters of our kernels to fixed high pass filters which may affect the generalization performance of model.

3.3.2 Convolution layers

As depicted in table 1, the proposed model uses 9 convolution layers. The first convolution layer “conv1” filters the 256×256×1 LOOP input image with 8 filters of size 7×7 using a stride of 1. The second convolution layer “conv2” filters the feature maps (250×250×8) produced by the first convolution layer with 8 kernels of 5×5 using a stride of 1. 246×246×8 feature maps output by the second convolution layer is filtered by the third convolution layer “conv3” with 16 kernels of size 5×5 yielding a feature map of 242×242×16. The first three convolution layers are connected directly without pooling layers to prevent information loss. These layers are responsible for extracting and learning median filtering fingerprints from the input images. The next six convolution layers namely “Conv4” with 16 filters of size 3×3×16 and stride of 1, “Conv5” with 32 filters of size 3×3 and stride of 1, “Conv6” with 32 filters of size 3×3 and stride of 1, “Conv7” with 64 filters of size 3×3 and stride of 1, “Conv8” with 64 filters of size 3×3 and stride of 1, “Conv9” with 128 filters of size 3×3 and stride of 1, which are responsible for learning further discriminative features yielded the following output dimensions, 120×120×16, 60×60×32, 30×30×32, 15×15×64, 13×13×64, and 13×13×128 respectively.

3.3.3 Batch Normalization (BN)

The distribution of input data to internal layers changes as data flows through deep neural networks during training due to changes in the network parameters. This affects the learning capacity and accuracy of the network. To partially overcome this problem, we applied BN after each regular convolution layer of the proposed CNN design. The BN after each convolution layer normalizes elements in each feature map to zero (0) mean and unit (1) variance before feeding it to the next layer while training, hence accelerating training and increasing overall accuracy.

3.3.4 Activation function

To introduce nonlinearity to neural networks, which greatly increases their capabilities of feature representation, a convolution layer is usually followed by a non-linear mapping known as an activation function. There is a variety of choice for an activation function such as tanh, sigmoid, and relu, etc. however, we tested the performance of the proposed model using various activation functions, combing more than one type of activation in some test cases, our investigation shows that the proposed model works best while using tanh activation. Similar to [21] and due to the fragile nature of feature maps extracted in early layers (1-3), our model only introduce activation functions after the third convolution layer. Layers 4-10 uses the tanh activation and the softmax activation is applied after the last fully-connected layer (layer 11), which is responsible for mapping features learned by the last FC layer to a set of probability values, each corresponding to one class in the n+1 classes under consideration.

3.3.5 Pooling

Pooling operation is often applied after regular convolution layer in CNN to reduce the dimension of the feature maps and to obtain a more compact representation of the input data. Thus, reducing computational burden and chances of over-fitting. Max pooling and average pooling are the most widely used pooling operations in CNN design. Max pooling does not take into account all the activations within the pooling region, instead, it returns only the strongest activation within the region. Such kind of activation works best for sparse feature representation [22].

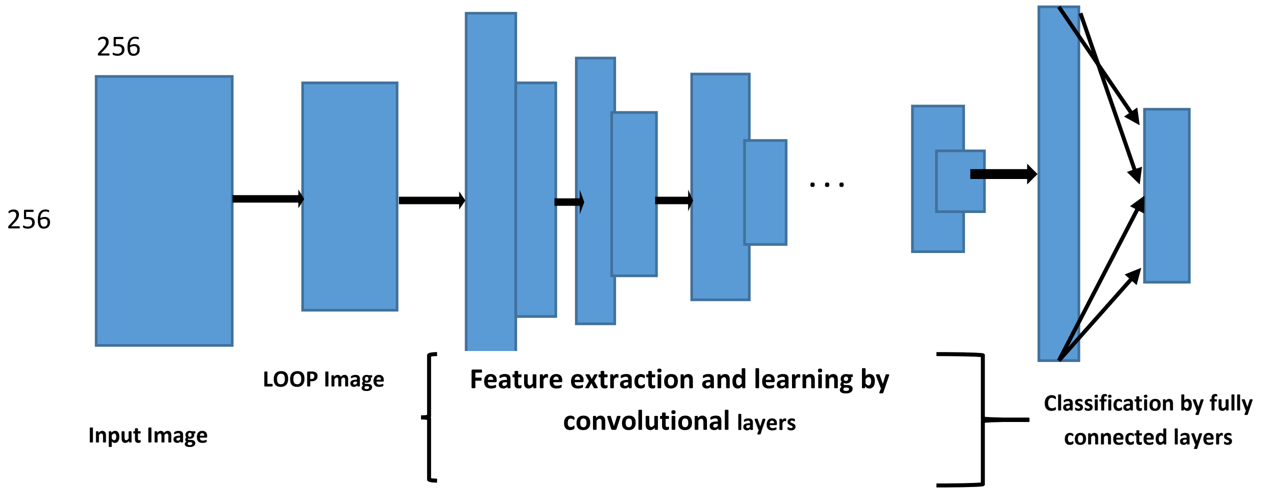

256
256
Input Image
LOOP Image
. . .
Feature extraction and learning by convolutional layers
Classification by fully connected layers

Figure 1. An overview of the proposed framework for median filtering detection. First, an input image is converted to LOOP representation which serves as input to the CNN network. Then the convolution layers of the network extracts and learns median filtering features from the loop images. Finally, the learned features are forwarded to the fully connected layer for classification

$$\begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \quad \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix} \quad \begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix} \quad \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{pmatrix}$$

North **North East** **East** **South East**

$$\begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix} \quad \begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix} \quad \begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix} \quad \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}$$

South **South West** **West** **North West**

(a)

250	200	50
245	150	40
20	15	10

(b)

3070	1510	-170
1630		-1690
-250	-2130	-2120

(c)

2^7 1	2^5 1	2^4 0
2^6 1		2^2 0
2^3 0	2^0 0	2^1 0

(d)

LOOP value = $2^7 + 2^6 + 2^5 + 0 = 224$

Figure 2. Numerical Example of LOOP computation. Figure 2(a) shows the eight Kirsch mask and their directions, figure2 (b), (c) and (d) show a 3x3 image window, its eight response values using the eight kirsch mask (M_n, n = 0,1,2,…,7) and the corresponding binary values of the eight kirsch mask response with their weights respectively.

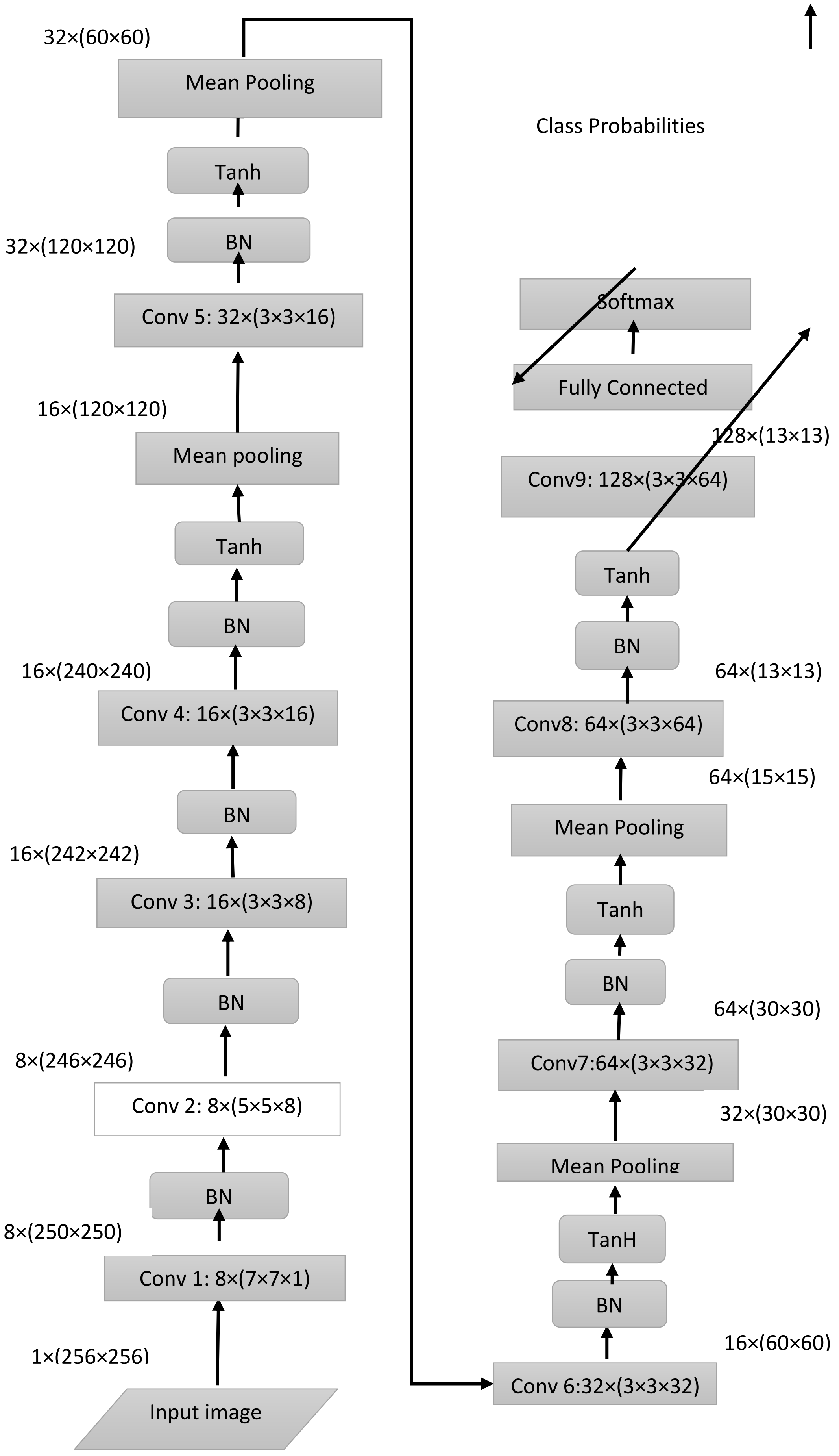


Figure 3. Proposed CNN Architecture

However, applying max-pooling in applications such as image forensics or steganalysis where the weak signals are the signal of interest may lead to the loss of the desired information. Therefore, throughout our proposed model, we used average pooling which takes into account all the activations within the pooling region. It is also worth pointing out that since pooling operations like non-linear activations are information losing processes [21], our CNN design delayed the introduction of pooling layers until after the fourth convolution layer as depicted in figure 3. Pooling operations suppresses the noise structures introduced by image manipulation operations, which are the signal of interest in image tampering detection, leading to poor performance

4. EXPERIMENTS

We evaluated the proposed model under different experimental setups (1) using the original image as input, (2) using LOOP images as input, in both compressed and uncompressed images with different resolutions. Lastly, the performance of the proposed model is compared with that of two states of the art methods from the literature using a composite datasets formed from five publicly available image databases.

4.1 Datasets

For a thorough assessment of the proposed method, 7,000 images formed from five publicly available image datasets, the UCID dataset [23], the Bossbases dataset [24], the Dresden image dataset [25], NRCS photo gallery dataset [26], and MS COCO dataset [27] were used to evaluate the proposed model. Each of these datasets contributes 1400 images to form the 7000 composite image datasets used for our experiments. To create the first set of the dataset for our experiments, each of these 7000 images was median filtered using 5x5 and 3x3 median filtering in uncompressed format for different resolution of images to obtain their positive classes, while the original images constitute the negative classes. The second set of the dataset is obtained by applying JPEG compression with different quality factors (q = 90, 30) to the first set of the dataset. The LOOP images for both sets of the dataset were then computed and used as input of the proposed model. 70% of the entire dataset were randomly selected for training, 10% for validation and the remaining 20% was kept for testing purposes.

4.2 Implementation details

All experiments were carried out using Tensorflow on a machine with NVidia GeForce GTX 1050 GPU and 8GB RAM. For training, we used Adam optimizer with a learning rate of 0.0001 and the default values for the moments ($\beta1 = 0.9$, $\beta2 = 0.999$, and $\varepsilon = 10^{-7}$). Xavier initialization was used to initialize the weights of layer 1 through 9 of the proposed model with their biases set to 0. The two fully connected layers are initialized with random numbers from a zero-mean Gaussian distribution with 0.01 standard deviation and their biases were set to 0.05. We trained the proposed model in each experiment for 60 epochs without shuffling of training data between epochs with a mini-batch of 16 images for each training iteration to minimize the cross-entropy loss function to find the best set of parameters for the network.

Suppose θ is the parameter vector representing the weight vector corresponding to the median filtering detection task, the cross-entropy loss can be computed as:

$$L(\theta) = \frac{-1}{M}\sum_{m=1}^{M}\sum_{n=1}^{N} 1(y^m = n)\log(y^m = n|x^m;\,\theta) \qquad (3)$$

Where M and N denote the total number of image samples and the number of classes, 1(.) represents an indicator function which equals 1 if m = n, otherwise 0. y^m and x^m correspond to the image label and the feature of the sample m.

We minimizing the cross-entropy loss using the Adam optimizer with all the training samples to learn the network's optimal set of parameters. Using these learned parameters, the network can predict whether a given image is median filtered or not from the test samples.

4.3 Results

.We use the minimum probability of error of classification (PE) and receiver operating characteristic (ROC) curve as evaluation metrics for comparing the performance of models. ROC curve measures the performance of a binary classification task by varying the threshold on prediction score. The area under the ROC curve (AUC) is computed from the ROC curve that measures the ability of a system for binary classification and allows comparison between different methods. PE is a binary evaluation metric that computes the minimum probability of error of a binary classifier as

$$PE = \min\left(\frac{p_{fp}+p_{fn}}{2}\right)$$

(4)

Where P_{fp} and P_{fn} denote the false positive and false negative rates respectively.

4.3.1 Median Filtering detection in uncompressed images using the original and LOOP image

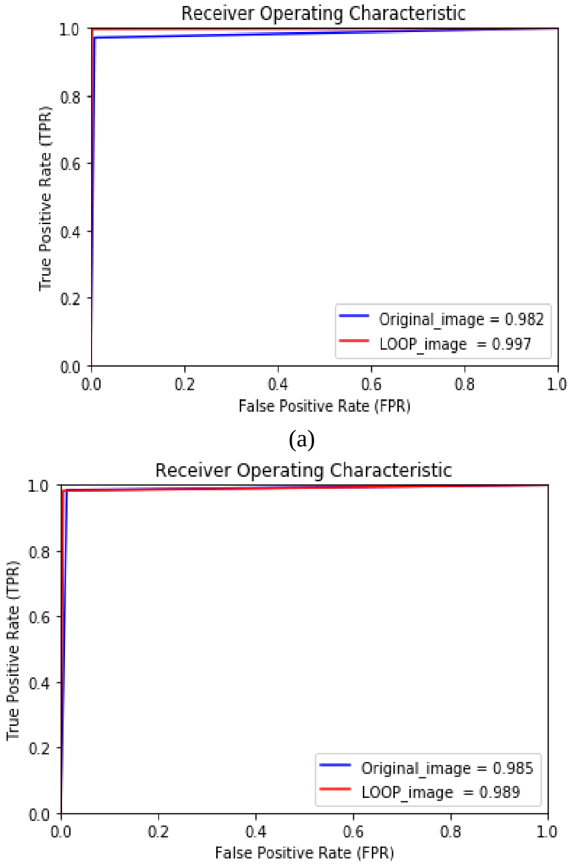
Tables 1 and 2 show the results in terms of PE and AUC for 5x5 and 3x3 median filtering respectively. Results of these experiments indicate that our model can achieve almost a perfect classification in both 5x5 and 3x3 median filtered images. From the results, we can observe that the proposed model can detect 5x5 median filtering with an AUC of 0.997 and a PE of 0.0029 in the uncompressed image and 3x3 median filtering with an AUC of 0.993 and a PE of 0.0065 in uncompressed images. Further, we can also observe that the use of LOOP images as input significantly improves the detection rate of the proposed method in both types of median filtering regardless of the image resolutions. Hence, these results demonstrate the effectiveness of the proposed method in detecting median filtering in uncompressed low-resolution images. Figures 4 and 5 further illustrate the performance of the proposed model in terms of ROC and corresponding AUC values for 5x5 and 3x3 median filtering respectively.

Table 1. Performance evaluation of the proposed method for 5x5 median filtering with varying image size using original and LOOP images

S = input size	Image	PE	AUC
256	Original	0.0179	0.982
	LOOP	**0.0029**	**0.997**
128	Original	0.0146	0.985
	LOOP	**0.0121**	**0.989**

Table 2. Performance evaluation of the proposed method for 3x3 median filtering with varying image size using original and LOOP images

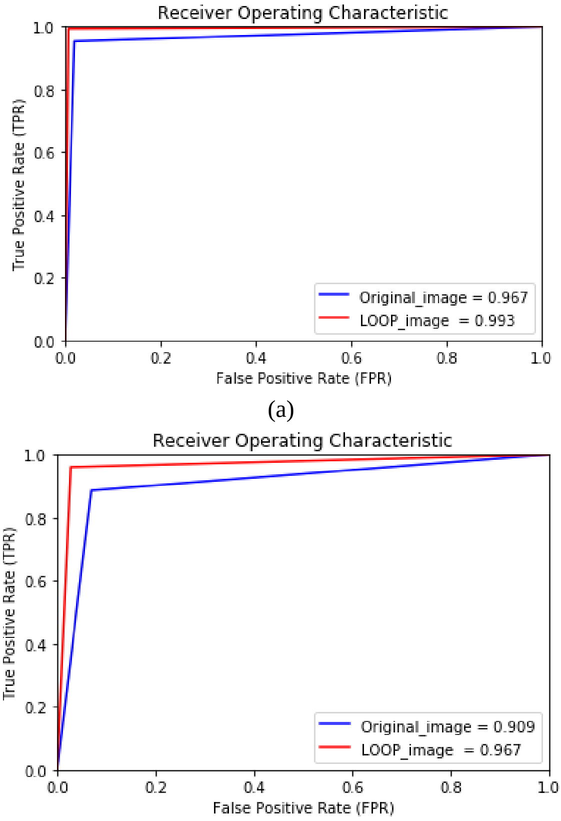
S = input size	Image	PE	AUC
256	Original	0.0321	0.967
	LOOP	**0.0065**	**0.993**
128	Original	0.0916	0.909
	LOOP	**0.0331**	**0.967**



(a)

(b)

Figure 4. ROC curves and their corresponding AUC values of the proposed method for 5x5 median filtering in uncompressed images for input sizes (a) 256 and (b) 128



(a)

(b)

Figure 5. ROC curves and their corresponding AUC values of the proposed method for 3x3 median filtering in uncompressed images for input sizes (a) 256 and (b) 128

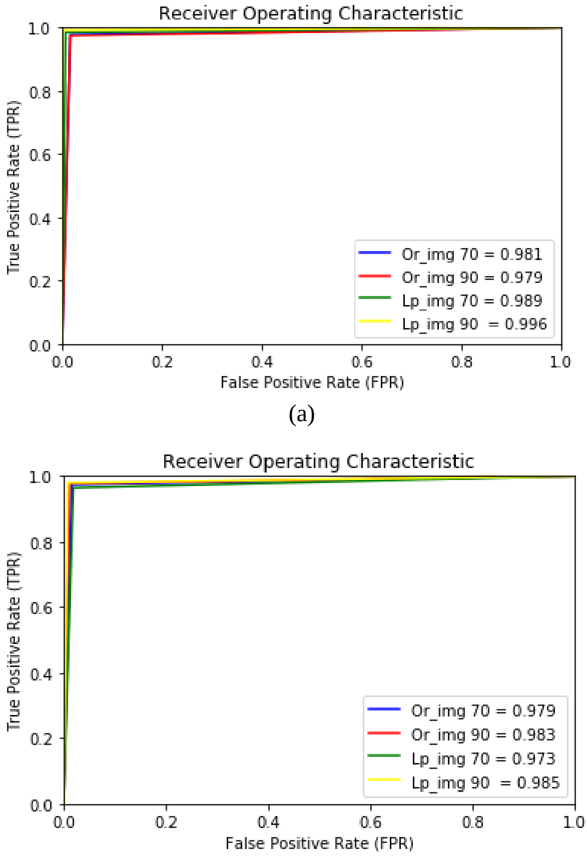
4.3.2 Median filtering detection in JPEG compressed images

To create a convincing image manipulation, forgers often applied JPEG compression after median filtering operation to hide traces of median filtering. Thus, a robust median filtering detector should be effective against post JPEG compression. Hence, we evaluate the proposed model for detecting median filtered images in compressed images using two different compression quality factors: (q = 90, and q = 70).

We denote the original and LOOP images with the compression quality factor applied as follows: Or_img90 and Or_img70 correspond to original median filtered images compressed with 90 and 70 quality factors respectively. Similarly, Lp_img90 and Lp_img70 correspond to the LOOP median filtered images compressed with 90 and 70 compression quality factors respectively. The results of the proposed model in JPEG compressed images (for q = 90 and 70) in terms of AUC and PE are presented in Tables 3 and 4 for 5x5 and 3x3 median filtering. From the results, it can observe that our model can achieve an AUC of 0.996 and PE of 0.0041 in 5x5 median filtering and an AUC of 0.989 and PE of 0.0108 in 3x3 median filtering while using the LOOP images as input. By contrast, our model achieves an AUC of 0.979 and PE of 0.0165 in 5x5 median filtering and AUC of 0.953 and PE of 0.0467 in 3x3 median filtering while using the original images. These results show that the proposed method can detect median filtered images in low-resolution images under JPEG compression. However, the performance is found best when using LOOP images as input as opposed to the original images. Thus, the proposed method also leads to some performance gain in terms of robustness against post JPEG compressed images. We also observed a little drop in performance of the proposed model in lower resolution images compressed with low-quality factors, especially for 3x3 median filtered images. Figures 6 and 7 further illustrate the performance of the proposed model in terms of ROC and corresponding AUC values for 5x5 and 3x3 median filtering in post JPEG compression respectively.

Table 3. Performance evaluation of the proposed method on JPEG compression for 5x5 median filtering with varying image size and compression quality

S = input size	Image	PE	AUC
256	Ori_img 70	0.0193	0.981
	Ori_img 90	0.0165	0.979
	Lp_img 70	0.0109	0.989
	Lp_img 90	**0.0041**	**0.996**
128	Ori_img 70	0.0212	0.979
	Ori_img 90	0.0188	0.983
	Lp_img 70	0.0273	0.973
	Lp_img 90	**0.0146**	**0.985**



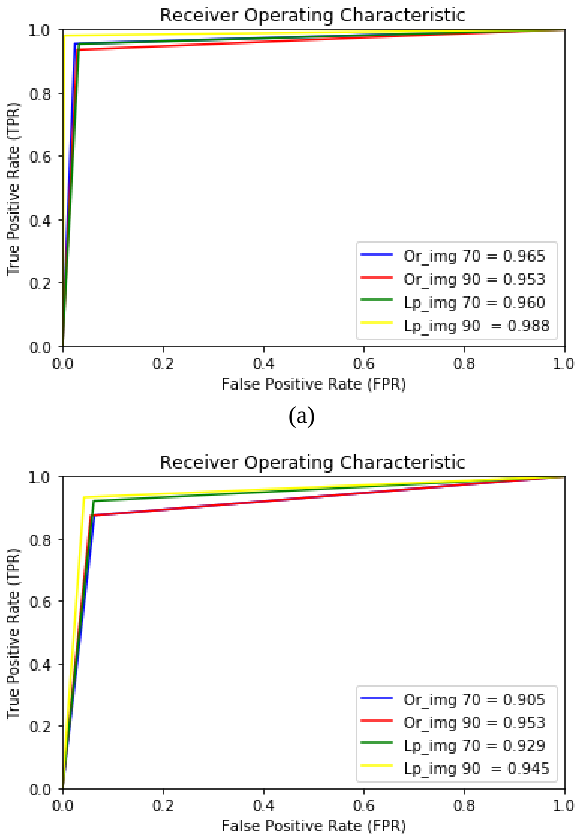
(a)

(b)

Figure 6. ROC curves and their corresponding AUC values of the proposed method for 5x5 median filtering in compressed images for input sizes (a) 256 and (b) 128

Table 4. Performance evaluation of the proposed method on JPEG compression for 3x3 median filtering with varying image size and compression quality

S = input size	Image	PE	AUC
256	Ori_img 70	0.0351	0.965
	Ori_img 90	0.0467	0.953
	Lp_img 70	0.0399	0.960
	Lp_img 90	**0.0108**	**0.989**
128	Ori_img 70	0.0947	0.905
	Ori_img 90	0.2004	0.908
	Lp_img 70	0.0708	0.929
	Lp_img 90	**0.0553**	**0.945**



(a)

(b)

Figure 7. ROC curves and their corresponding AUC values of the proposed method for 5x5 median filtering in compressed images for input sizes (a) 256 and (b) 128

4.3.3 Comparison against state of the art from the literature

To validate the performance of the proposed method, we compare the results obtained from our experiments with that of Chen [8] and Gupta [5] using the same datasets. The results are reported only in terms of PE under the uncompressed (UNC), 90 and 70 compression Quality Factors (QF) for 5x5 and 3x3 median filtered image as shown in Tables 5 and 6. The results of these experiments showed that our method outperforms both [8] and [5] in detecting median filtered images with low QF (QF = 90, 70). By contrast, the method of [5] outperformed the proposed method under the uncompressed (UNC) median filtered images. This is because our approach can learn better salient median filtering fingerprints directly from input images than the existing approach, which allows it to extract better discriminative features. Thus, these results demonstrate that the proposed method can effectively detect median filtering in both uncompressed and median filtered images compressed with low-quality factors.

Table 5. Comparison against handcrafted and CNN based methods for 5x5 median filtering

QF	Proposed	Chen [8]	Gupta [5]
UNC	0.0029	0.0057	**0.0008**
90	**0.0041**	0.0250	0.0058
70	**0.0109**	0.0422	0.0168

Table 6. Comparison against handcrafted and CNN based methods for 3x3 median filtering

QF	Proposed	Chen [8]	Gupta [5]
UNC	0.0070	0.0032	**0.0004**
90	**0.0108**	0.0871	0.0223
70	**0.0399**	0.0858	0.0509

5. CONCLUSION

In this paper, a novel method for detecting median filtering based on LOOP and CNN has been proposed. The proposed method addresses the challenge of detecting median filtering in low-resolution images compressed with low-quality factors for both 5x5 and 3x3 median filtered images. In the proposed method, the LOOP of the input images is computed and fed to the designed CNN from which deep features are extracted and used for classification. Extensive experiments show that the proposed method can effectively detect median filtering in low-resolution images in both uncompressed and compressed format. In the future, we plan to improve the proposed model to a general-purpose image manipulation detection framework that will detect both contents changing manipulations and other content preserving manipulations.

6. ACKNOWLEDGMENTS

We sincerely thank the National Information Technology Development Agency (NITDA) of Nigeria, for funding this research.

7. REFERENCES

[1] Kakar P. Passive Approaches for detecting digital image forgery. *Ph.D. Thesis*, Nanyang Technological University, 2012.

[2] Tyagi V. Detection of forgery in images stored in digital form. *Project report submitted to DRDO, New Delhi,* 2010.

[3] Farid H. A survey of image forgery detection. *IEEE Signal Process. Mag*., 2009, Vol. 26, no. 2, pp. 1625.

[4] Kirchner M. Fridrich J. On detection of median filtering in digital images. *Proc. SPIE*, 2010, vol. 7541, Art. no. 754110.

[5] Gupta A, Singhal D. Global median filtering forensic method based on Pearson parameter statistics. *IET Image Processing*, 2019 Mar 29; 13(12):2045-57.

[6] Gupta A, Singhal D. A Simplistic Global Median Filtering Forensics Based on Frequency Domain Analysis of Image Residuals. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2019 Aug 20; 15(3):1-23.

[7] Tang H, Ni R, Zhao Y, Li X. Median filtering detection of small-size image based on CNN. *Journal of Visual Communication and Image Representation*, 2018 Feb 1; 51:162-8.

[8] Chen J, Kang X, Liu Y, Wang ZJ. Median filtering forensics based on convolutional neural networks. *IEEE Signal Processing Letters*, 2015 Jun 1; 22(11):1849-53.

[9] Yu L, Zhang Y, Han H, Zhang L, Wu F. Robust median filtering forensics by CNN-based multiple residuals learning. *IEEE Access*, 2019 Aug 2, 7:120594-602.

[10] Chakraborti T, McCane B, Mills S, Pal U. Loop descriptor: Local optimal-oriented pattern. *IEEE Signal Processing Letters*, (2018), *25*(5), pp.635-639.

[11] Cao G, Zhao Y, Ni R, Yu L, Tian H. Forensic detection of median filtering in digital images. *In Proc. IEEE Conf. Multimedia Expo,* Jul. 2010, pp. 89–94.

[12] Yuan HD. Blind forensics of median filtering in digital images. *IEEE Transactions on Information Forensics and Security*. 2011 Jul 14; 6(4):1335-45.

[13] Niu Y, Zhao Y, Ni R. Robust median filtering detection based on local difference descriptor. *Signal Processing: Image Communication*, 2017 Apr 1; 53:65-72.

[14] Gupta A, Singhal D. Analytical global median filtering forensics based on moment histograms. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM),* 2018 Apr 25; 14(2):1-23.

[15] Farooq S, Yousaf M. H, Hussain F. A generic passive image forgery detection scheme using local binary pattern with rich models. *Computers & Electrical Engineering*. (2017), 62, pp.459-472.

[16] Ojala T, Pietikinen M, Harwood D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. *In Proc. ICPR*, 1994.

[17] Jabid T, Kabir M. H, Chae O.S. Gender Classification using Local Directional Pattern (LDP). *In Proc. ICPR,* 2010.

[18] Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J, Chen T. Recent advances in convolutional neural networks. *Pattern Recognition*, 2018, 77, pp.354-377.

[19] Bayar B, Stamm M. C. A deep learning approach to universal image manipulation detection using a new

convolutional layer. *In Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security,* (2016), 5 -10.

[20] Bayar B, Stamm M C. Constrained convolutional neural networks: A new approach towards general-purpose image manipulation detection. *IEEE Trans. Inf. Forensics Security*, 2018, vol. 13, no. 11, pp. 2691–2706.

[21] Boroumand M, Fridrich J. Deep learning for detecting processing history of images. *Society for Imaging Science and Technology,* (2018).

[22] Qian Y, Dong J, Wang W, Tan T. Deep learning for steganalysis via convolutional neural networks. *In Media Watermarking, Security, and Forensics, 2015,* (Vol. 9409, p. 94090J).

[23] Schaefer, G., Stich, M.: 'Ucid: an uncompressed color image database'. Proc. SPIE, San Jose, California, 2003, vol. 5307, pp. 472–480. Available at http:// dx.doi.org/10.1117/12.525375

[24] Bas P, Filler T, Pevny T. Break our steganographic system:The ins and outs of organizing boss. *In Information Hiding*, 2011, pp. 59-70.

[25] Gole T, Bohme R. Dresden Image Database for benchmarking digital image forensics. *In Proceedings of the 2010 ACM Symposium on Applied Computing*, March 22-26, (2010).

[26] NRCS, U: 'Natural resources conservation service photo gallery, United States department of agriculture', 2014. Available at http://plants.usda.gov/

[27] Lin T Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, *et al.*. Microsoft COCO: Common objects in context. *In ECCV*, 2014, 2- 5.